ELSEVIER

Theoretical Computer Science 266 (2001) 935–950

Theoretical Computer Science

www.elsevier.com/locate/tcs

Underspecification for a simple process algebra of recursive processes

M.E. Majster-Cederbaum *

Institut für Informatik, Fak. für Mathematik und Informatik, Universität Mannheim, Lehrstuhl Prakt. Informatik II, D27, 27, 68131 Mannheim, Germany

Received March 1999; revised September 2000; accepted November 2000
Communicated by M. Nivat

Abstract

This paper deals with underspecification for process algebras which is relevant in early design stages. We consider a form of underspecification that arises from a situation where at a certain design stage the decision between several options of system behaviour is to be postponed until more information is available. We follow an approach of Veglioni and De Nicola (Lecture Notes in Computer Science 1466 (1998) 179) who propose to interpret the choice operator $+$ of a simple class of finite process terms as underspecification whenever it combines two processes that have some initial action in common, as e.g. in $(a.P + b.Q) + (a.R + c.S)$. In particular, we consider recursive processes and discuss several extensions.

Keywords: Specification; Underspecification; Semantics; Refinement

1. Introduction

During the design of a reactive system, it may occur that certain information relevant for the present stage of the specification is either not yet available or should be supressed but will be provided at a later stage. This is where the concept of underspecification comes in. One way to handle the situation of incomplete information is to admit a specification that leaves room for later refinement steps in which additional knowledge can be incorporated.

In a recent paper [10] Veglioni and De Nicola considered a simple process algebra **BP** of finite processes and proposed to use the operator $+$ as underspecification

* Tel.: +49-621-2923; fax: +49-621-2925.
E-mail address: mcb@informatik.uni-mannheim.de (M.E. Majster-Cederbaum).

0304-3975/01/$ - see front matter
PII: S0304-3975(00)00419-9

whenever it combines processes which have some initial action in common. The meaning of such a term is then a set of deterministic trees (possible worlds) where each tree represents one option of the specification. Refinement can then be modelled by inclusion between sets of possible worlds and induces a relation on process terms that is weaker than bisimulation. We show here how these ideas carry over to infinite (recursive) processes by using a metric setting while preserving the relation between bisimulation and refinement. We discuss the introduction of further operators and operational aspects.

The paper is organized as follows. Section 2 contains the definitions. In Section 3 we describe a domain of trees upon which we will build our semantics. Section 4 presents the denotational possible worlds semantics for the language of recursive processes. In Section 5 properties of the possible worlds semantics are presented. In Section 6 we discuss several extensions and related work.

2. Definitions

In the following, we give a summary of the concepts of processes and some notions from metric topology. We consider processes that are able to perform *actions* from a given set **Act**. An action represents any activity of a system at a chosen level of abstraction. A *transition system over* **Act** is a pair $(\mathbf{A}, \rightarrow)$, where $\mathbf{A}$ is the class of processes (or states) and $\rightarrow \subseteq \mathbf{A} \times \mathbf{Act} \times \mathbf{A}$ is the transition relation. We write $p \xrightarrow{a} q$ for $(p,a,q) \in \rightarrow$. On transition systems a variety of semantic equivalences have been investigated as e.g. presented in [9].

We first extend the class **BP** of [10] of finite processes by recursion to model infinite behaviour. A parallel construct, concatenation and infinite summation are discussed in Section 6. The class **RBP** of processes is given by

- $0 \in \mathbf{RBP}$
- $a.P \in \mathbf{RBP}$ (prefix) for all $a \in \mathbf{Act}$, $P \in \mathbf{RBP}$
- $X \in \mathbf{RBP}$ for all $X \in Idf$
- $P + Q \in \mathbf{RBP}$ (sum) for all $P, Q \in \mathbf{RBP}$
- $fix(X = P)$ for all $X \in Idf, P \in \mathbf{RBP}$ such that X is guarded in P.

Here Idf is a set of identifiers. An occurrence of $X \in Idf$ is *free* in P iff it does not occur within a subterm of the form $fix(X = Q)$. An identifier $X \in Idf$ is *guarded* in P iff each free occurrence of X in P is in the scope of a prefix operation. A process is *closed* iff it does not contain any free occurrences of identifiers. For $P, Q \in \mathbf{RBP}$, $X \in Idf$, $P[X/Q]$ denotes the process where each free occurrence of X in P is substituted by Q. **RBP** yields a labelled transition system with the transitions $a.P \xrightarrow{a} P$, $P + Q \xrightarrow{a} P'$ if $P \xrightarrow{a} P'$ or $Q \xrightarrow{a} P'$, $fix(X = P) \xrightarrow{a} P'$ if $P[X/fix(X = P)] \xrightarrow{a} P'$.

We want to define the subclass of deterministic processes. For this, we introduce a set of assignments that associate a set of actions with each identifier $INIT = \{\sigma \mid \sigma : Idf \rightarrow \mathcal{P}(\mathbf{Act})\}$, and a function $I : \mathbf{RBP} \rightarrow (INIT \rightarrow \mathcal{P}(\mathbf{Act}))$ giving the set of initial actions

for a process P defined as follows:

$I(0)(\sigma)=\emptyset$ $\qquad$ $I(X)(\sigma)=\sigma(X)$

$I(a.P)(\sigma)=\{a\}$ $\qquad$ $I(P+Q)(\sigma)=I(P)(\sigma)\cup I(Q)(\sigma)$

$I(\mathit{fix}(X=P))(\sigma)=\mathit{lf}\,\Phi_{P,X}(\sigma)$,

where $\mathit{lf}\,\Phi_{P,X}(\sigma)$ is the least fixed point of $\Phi_{P,X}(\sigma):\mathscr{P}(\mathbf{Act})\to\mathscr{P}(\mathbf{Act})$ given by $\Phi_{P,X}(\sigma)(U)=I(P)(\sigma)[X/U]$.

Let $P\in\mathbf{RBP}$, $\sigma\in INIT$. The relation $\delta\subset\mathbf{RBP}\times INIT$ characterizes the processes that are deterministic under an assignment σ and is given as follows:

$(0,\sigma)\in\delta$

$(X,\sigma)\in\delta$

$(a.P,\sigma)\in\delta$ if $(P,\sigma)\in\delta$

$(P_1+P_2,\sigma)\in\delta$ if $I(P_1)\sigma\cap I(P_2)\sigma=\emptyset$ and $(P_i,\sigma)\in\delta$ $i=1,2$

$(\mathit{fix}(X=P),\sigma)\in\delta$ if $(P,\sigma)\in\delta$

$(P,\sigma)\in\delta$ means that P is deterministic under σ. Let us call a process P *deterministic* if $(P,\sigma)\in\delta$ for all σ.

Example 1. 0, X, $\mathit{fix}(X=a.X+b.Y)$ are deterministic; if P,Q are deterministic, so is $a.P+b.Q$; $\mathit{fix}(X=a.X+Y)$ is not deterministic.

To handle recursion, we will use the metric setting, first proposed by [7] and investigated in [3, 6]. We refer to [4] for basic facts from (metric) topology.

If (M,d_M) is a (complete) metric space then $(\mathscr{P}_{nco}(M),d_H)$ is a (complete) metric space where

$$\mathscr{P}_{nco}(M)=\{U\subseteq M\mid U\neq\emptyset,\ U\ \text{compact}\}$$

and

$$d_H(X,Y)=\max\left\{\sup_{x\in X}\inf_{y\in Y}d(x,y),\sup_{y\in Y}\inf_{x\in X}d(x,y)\right\}$$

for $X,Y\in\mathscr{P}_{nco}(M)$.

If (M,d_M) is a discrete space then $\mathscr{P}_{nco}(M)=\mathscr{P}_{nf}(M)$, where

$$\mathscr{P}_{nf}(M)=\{U\subseteq M\mid U\neq\emptyset,\ U\ \text{finite}\}$$

Theorem 1. *Let (M,d) be a metric space. If $X\subseteq\mathscr{P}_{nco}(M)$ is compact and $X\neq\emptyset$ then $\bigcup_{A\in X}A\in\mathscr{P}_{nco}(M)$.*

Proof. Let U be an open cover for $\bigcup A$ and $A\in X$. As A is compact there must be a finite subset of U that covers A and yields an open neighbourhood $U(A)$ of A. Hence

$\{U(A)\}_{A\in X}$ is an open cover for X, from where we obtain a finite cover of X, as X is compact. From this finite cover we obtain a finite cover for $\bigcup A$. □

The next theorem will enable us to lift certain set-valued operations f defined on trees to operations defined on compact sets of trees by pointwise application of f. As the theorem is stated for arbitrary metric spaces it can also serve as a basis for a possible worlds semantics that is based on other models than trees.

Theorem 2. *Let $(M,d_M),(N,d_N)$ be metric spaces. $f:M\times M\to\mathscr{P}_{nco}(N)$ a non-distance-increasing function. We put for $U,V\in\mathscr{P}_{nco}(M)$*

$$\hat{f}(U,V)=\bigcup_{u\in U,v\in V} f(u,v)$$

then

(i) *$\hat{f}(U,V)$ is a nonempty compact subset of N for all $U,V\in\mathscr{P}_{nco}(M)$.*
(ii) *$d_H(\hat{f}(U,V),\hat{f}(U',V'))\leqslant\max(d_H(U,U'),d_H(V,V'))$ for all $U,V\in\mathscr{P}_{nco}(M)$, i.e. $\hat{f}:\mathscr{P}_{nco}(M)\times\mathscr{P}_{nco}(M)\to\mathscr{P}_{nco}(N)$ is non-distance-increasing.*

Proof. (i) We first observe that $S=\{f(u,v)|u\in U,\ v\in V\}$ is a nonempty compact set for $U,V\in\mathscr{P}_{nco}(M)$: let $(f(u_i,v_i))_{i\in I}$ be a sequence in S; hence, $((u_i,v_i))_{i\in I}$ is a sequence in $U\times V$, hence there is a subsequence $((u_{i_j},v_{i_j}))_{j\in J}$ of $((u_i,v_i))_{i\in I}$ that converges to some (u_0,v_0) in $U\times V$. As f is non-distance-increasing $(f(u_{i_j},v_{i_j}))_{j\in J}$ converges to $f(u_0,v_0)\in S$. Application of Theorem 1 yields the result.

(ii) Let $U,V\in\mathscr{P}_{nco}(M)$. We first observe that

$$\begin{aligned}
&d(\hat{f}(U,V),\hat{f}(U',V'))\\
&\quad\leqslant d(\{f(u,v):\ u\in U,\ v\in V\},\{f(u',v'):\ u'\in U',v'\in V'\})\\
&\quad=\max\Bigg(\sup_{u\in U,v\in V}\inf_{u'\in U',v'\in V'} d(f(u,v),f(u',v')),\\
&\qquad\sup_{u'\in U',v'\in V'}\inf_{u\in U,v\in V} d(f(u,v),f(u',v'))\Bigg)\\
&\quad\leqslant\max(d(U,U'),d(V,V')).
\end{aligned}$$

3. A domain of trees

The possible worlds semantics of **BP** is given in [10] in terms of finite sets of finite deterministic trees with edge labels in **Act**. In order to be able to model the meaning of recursive processes we define in the following a suitable metric space $(\mathbf{D},d)$ of trees and use $\mathscr{P}_{nco}(\mathbf{D})$ as semantic domain for **RBP**, as compactness generalizes finiteness. The choice of **D** is justified as follows. Bisimular terms in **BP** obtain the same meaning

in [10], hence this semantics can be viewed as a mapping from **BP** to $\mathscr{P}_{nf}(\mathbf{T}_{\text{finbran}}/_{\sim})$. Here $\mathbf{T}_{\text{finbran}}$ denotes the class of (isomorphism classes of) finitely branching trees with edge labels in **Act** and $\sim$ denotes bisimulation. The natural metric on $\mathbf{T}_{\text{finbran}}$ is given by

$$d_T(t_1,t_2)=\inf\left\{\frac{1}{2^n}|t_1^{(n)}=t_2^{(n)}\right\},$$

where $t^{(n)}$ denotes the n-cut of t. With this metric $\mathbf{T}_{\text{finbran}}$ is a complete metric space. The metric carries over to $\mathbf{T}_{\text{finbran}}/_{\sim}$ and yields an incomplete metric space $(\mathbf{T}_{\text{finbran}}/_{\sim}, d_T)$. Let (Δ,δ) denote the completion of $(\mathbf{T}_{\text{finbran}}/_{\sim}, d_T)$. (Δ,δ) can be given an alternative, more flexible characterization as follows. Let **CMS** be the category where the objects are complete metric spaces and the arrows are non-distance-increasing functions. The functor $\mathscr{F}:\mathbf{CMS}\to\mathbf{CMS}$ is given by

$$\mathscr{F}(M)=\{\emptyset\}\cup\mathscr{P}_{nco}(\mathbf{Act}\times M)$$

and

$$\mathscr{F}(f)=\lambda U.\{(a,f(m))\,|\,(a,m)\in U\}.$$

It is a well-known fact that $\mathscr{F}$ has a unique fixed point in **CMS** [6, 3] that can be obtained as the metric completion $(\mathbf{D},d)$ of $\bigcup_{i\geqslant 0}D_i$, where

$$D_0=\{\emptyset\},\quad D_{i+1}=\mathscr{F}(D_i)\quad i\geqslant 0.$$

As D_i is discrete for $i\geqslant 0$, we have

$$D_{i+1}=\{\emptyset\}\cup\mathscr{P}_{nf}(\mathbf{A}\times D_i)\quad i\geqslant 0.$$

$\bigcup_{i\geqslant 0}D_i$ consists of finitely branching trees of finite height.

Theorem 3. *$(\mathbf{D},d)$ and (Δ,δ) are isometric.*

Proof. This proof consists of the following observations:
1. the mappings $F_n:\mathbf{T}^{(n)}_{\text{finbran}}/_{\sim}\to D_n$, defined by

$$F_n([t_\emptyset]_\sim)=\emptyset\quad\text{and}\quad F_n([t]_\sim)=\{(a,F_{n-1}([t']_\sim))\colon t\xrightarrow{a}t'\}$$

for

$$t\in\mathbf{T}^{(n)}_{\text{finbran}}=\{t\in\mathbf{T}_{\text{finbran}}\,|\,1\leqslant height(t)\leqslant n\}$$

are well defined, bijective and distance-preserving,
2. for every $t\in\mathbf{T}_{\text{finbran}}$ the sequence $F([t^{(n)}]_\sim)_{n\geqslant 0}$ is a Cauchy sequence in **D**,
3. the mapping $F:\mathbf{T}_{\text{finbran}}/_{\sim}\to\mathbf{D}$ given by $F([t]_\sim)=\lim_{n\to\infty}F_n([t^{(n)}]_\sim)$ is an embedding,
4. $F(\mathbf{T}_{\text{finbran}}/_{\sim})$ is dense in **D**. □

By standard arguments [3] one can introduce operators $\circ$, $+$, and $\cdot$ on $\mathbf{D}$ as follows:
$\circ$: corresponds to the empty tree $\emptyset$

$+\colon \bigcup_{i\geqslant 0} D_i \times \bigcup_{i\geqslant 0} D_i \to \bigcup_{i\geqslant 0} D_i,\ t_1 + t_2 := t_1 \cup t_2,$

$\cdot : \mathbf{Act} \times \bigcup_{i\geqslant 0} D_i \to \bigcup_{i\geqslant 0} D_i,\ a \cdot t := \{(a,t)\}$

As $+$ and $\cdot$ are non-distance-increasing on $\bigcup_{i\geqslant 0} D_i$ they may hence be uniquely extended to $\mathbf{D}$. The initial actions function I for trees in $\mathbf{D}$ is given by

$$I(t) = \left\{ a\colon (a,x) \in t \text{ for some } x \in \bigcup_{i\geqslant 0} D_i \right\} \quad \text{for } t \in \bigcup_{i\geqslant 0} D_i.$$

For $t = \lim t_n \in \mathbf{D}$ we choose some $\varepsilon < \frac{1}{2}$ and determine N such that $d(t_n, t) < \frac{1}{2}$ for all $n \geqslant N$. We put

$$I(t) = \bigcup_{k\geqslant N} I(t_k)$$

Lemma 1. *$I(t)$ is finite for all $t \in \mathbf{D}$.*

A tree in $\mathbf{D}$ is deterministic at node n if no two edges with the same label leave the node. It is called *deterministic* if this condition holds for every node.

4. Denotational infinite possible worlds semantics for RBP

In this section, we present the denotational semantics for **RBP** in terms of compact sets of deterministic trees where each tree represents one option of the specification at the present stage. The choice between the options is to be decided in a later design step thus performing a refinement.

For the definition of $\hat{*}$ that models the desired interpretation of $+$ we proceed as follows. We first define a function *rdet* that decomposes a tree t into a set $rdet(t)$ of root deterministic trees t' which have the same initial actions as t and then define an operation $*$ on root deterministic trees that already reflects the desired interpretation of $+$. This operation is extended to arbitrary trees by using the function *rdet* and then lifted to compact sets of trees using Theorem 2.

Remark 1. It should be noted that the corresponding operator $\bigstar$ of [10] *cannot* be used for trees in $\mathbf{D}$. This is due to the fact that $\mathbf{D}$ contains infinitely branching trees for which $\bigstar$ is not welldefined. However, $\bigstar$ and $*$ coincide for (finite sets of) finite deterministic trees and hence finite processes obtain the same meaning in both semantics.

Definition 1. Let $t \in \bigcup_{i\geqslant 0} D_i$. We put
$rdet(\emptyset) = \{\emptyset\}$. For $t \neq \emptyset$ let

$$F_t = \left\{ f \mid f\colon I(t) \to \bigcup_{i\geqslant 0} D_i \text{ such that } (a, f(a)) \in t \right\}$$

and

$$rdet(t) = \{\{(a, f(a)) \mid a \in I(t)\} \mid f \in F_t\}.$$

Remark 2. $rdet(t)$ is finite for all $t \in \bigcup_{i\geqslant 0} D_i$, $t = \bigcup_{t' \in rdet(t)} t'$ for all $t \in \bigcup_{i\geqslant 0} D_i$ and $rdet(t) = \{t\}$ for each root deterministic $t \in \bigcup_{i\geqslant 0} D_i$.

rdet decomposes a tree in $\bigcup_{i\geqslant 0} D_i$ into a set of root deterministic trees t' with $I(t) = I(t')$.

Lemma 2. *rdet*: $\bigcup_{i\geqslant 0} D_i \to \mathscr{P}_{nco}(\mathbf{D}) \cup \{\{\emptyset\}\}$ *is non-distance-increasing.*

Proof. By straightforward calculation. □

Hence *rdet* can be canonically extended to a non-distance-increasing map $rdet : \mathbf{D} \to \mathscr{P}_{nco}(\mathbf{D})$.

Definition 2. Let $t, t' \in \bigcup_{i\geqslant 0} D_i$ be root deterministic. Let

$$F_{t,t'} = \left\{ f \mid f : I(t) \cup I(t') \to \bigcup_{i\geqslant 0} D_i \text{ such that } (a, f(a)) \in t \vee (a, f(a)) \in t' \right\}.$$

We put

$$t * t' = \{\{(a, f(a)) \mid a \in I(t) \cup I(t')\} \mid f \in F_{t,t'}\}$$

For $t_1, t_2 \in \bigcup_{i\geqslant 0} D_i$ we put

$$t_1 * t_2 = \bigcup_{\substack{t \in rdet(t_1) \\ t' \in rdet(t_2)}} t * t'.$$

Lemma 3. *Let $t_1, t_2 \in \bigcup_{i\geqslant 0} D_i$ be deterministic trees. Then $t_1 * t_2$ contains only deterministic trees.*

Proof. By induction on the size of $I(t_1) \cap I(t_2)$. □

Lemma 4. *Let $t_1, t_2 \in \bigcup_{i\geqslant 0} D_i$ be root deterministic then*

$$d_H(t_1 * t_2, t_1' * t_2') \leqslant \max(d_H(t_1, t_1'), d_H(t_2, t_2')).$$

Proof. The case that $d_H(t_1, t_1') = 1$ or $d_H(t_2, t_2') = 1$ is trivial. Let now $d_H(t_1, t_1') < 1$ and $d_H(t_2, t_2') < 1$. Then $I(t_1) = I(t_1')$ and $I(t_2) = I(t_2')$. Let $T = t_1 * t_2$, $T' = t_1' * t_2'$. Each $t \in T$ is a combination of parts of t_1 and parts of t_2, i.e.

$$t = \{(a_i, \tau_i) \mid i \in I\} \cup \{(b_j, \tau_j) \mid j \in J\},$$

where $(a_i, \tau_i) \in t_1$, $(b_j, \tau_j) \in t_2$.

For each $i \in I$ choose τ_i' such that $(a_i, \tau_i') \in t_1'$ and for each $j \in J$ choose τ_j' such that $(b_j, \tau_j') \in t_2$ and put

$$t' = \{(a_i, \tau_i') \mid i \in I\} \cup \{(b_j, \tau_j') \mid j \in J\}.$$

Then

$$\sup_{t \in T} \inf_{t' \in T'} d_H(t, t') \leqslant \max(d_H(t_1, t_1'), d_H(t_2, t_2')).$$

Theorem 4. *Let* $t_1, t_2, t_1', t_2' \in \bigcup_{i \geqslant 0} D_i$, *then*

$$d_H(t_1 * t_2, t_1' * t_2') \leqslant \max(d_H(t_1, t_1'), d_H(t_2, t_2')).$$

Proof.

$$\begin{aligned} & d_H(t_1 * t_2, t_1' * t_2') \\ & = d_H\left(\bigcup_{\substack{t \in rdet(t_1) \\ \hat{t} \in rdet(t_2)}} t * \hat{t}, \bigcup_{\substack{t' \in rdet(t_1') \\ \hat{t}' \in rdet(t_2')}} t' * \hat{t}' \right) \\ & \leqslant \max(d_H(rdet(t_1), rdet(t_1')), d_H(rdet(t_2), rdet(t_2'))) \\ & \leqslant \max(d_H(t_1, t_1'), d_H(t_2, t_2')) \end{aligned}$$

as *rdet* is non-distance-increasing and as $*$ is a non-distance-increasing function on finite root deterministic trees by Lemma 4, and by applying Theorem 2. □

Hence $*$ can be canonically extended to a non-distance-increasing map

$$* : \mathbf{D} \times \mathbf{D} \to \mathscr{P}_{nco}(\mathbf{D}).$$

Theorem 5. *Let* T_1, T_2 *be nonempty compact sets of trees in* **D** *then*
(i) $T_1 \hat{*} T_2 := \bigcup_{t_i \in T_i} t_1 * t_2$ *is a nonempty compact subset of* **D**
(ii) $\hat{*} : \mathscr{P}_{nco}(\mathbf{D}) \times \mathscr{P}_{nco}(\mathbf{D}) \to \mathscr{P}_{nco}(\mathbf{D})$ *a non-distance-increasing function.*

Proof. By Theorems 2 and 4. □

Remark 3. If T_1, T_2 consist of deterministic trees then $T_1 \hat{*} T_2$ consists of deterministic trees.

Definition 3. Let $\mathbf{ENV} = \{\sigma \mid \sigma : Idf \to \mathscr{P}_{nco}(\mathbf{D})\}$ be the set of environments. For $T \in \mathscr{P}_{nco}(\mathbf{D})$, $X, Y \in Idf$

$$\sigma[X/T](Y) := \begin{cases} \sigma(Y), & Y \neq X, \\ T, & Y = X. \end{cases}$$

The meaning function $\langle\langle\cdot\rangle\rangle : \mathbf{RBP} \to \mathbf{ENV} \to \mathscr{P}_{nco}(\mathbf{D})$ is given by

$$\langle\langle 0\rangle\rangle(\sigma) = \{\emptyset\},$$

$$\langle\langle X\rangle\rangle(\sigma) = \sigma(X),$$

$$\langle\langle a.P\rangle\rangle(\sigma) = \{\{(a,t)\} \mid t \in \langle\langle P\rangle\rangle(\sigma)\},$$

$$\langle\langle P_1 + P_2\rangle\rangle(\sigma) = \langle\langle P_1\rangle\rangle(\sigma) \hat{*} \langle\langle P_2\rangle\rangle(\sigma),$$

$$\langle\langle fix(X = P)\rangle\rangle(\sigma) = fix\ \Phi_{P,X}(\sigma),$$

where $fix\Phi_{P,X}(\sigma)$ is the unique fixed point of the contractive mapping

$$\Phi_{P,X}(\sigma) : \mathscr{P}_{nco}(\mathbf{D}) \to \mathscr{P}_{nco}(\mathbf{D})$$

defined by

$$\Phi_{P,X}(\sigma)(T) = \langle\langle P\rangle\rangle\sigma[X/T].$$

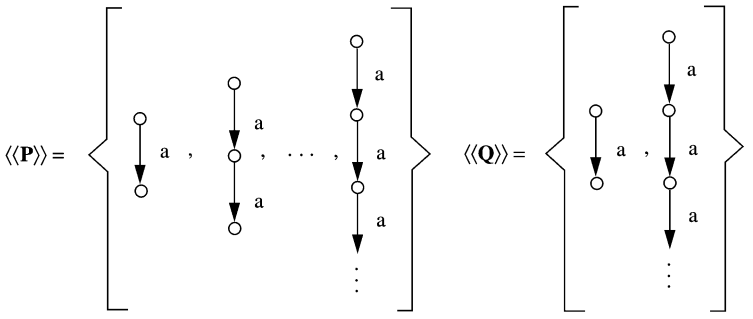
Remark 4. For each closed process P, $\langle\langle P\rangle\rangle$ is a set of deterministic trees in **D**.

Example 2. Let $P = fix(X = a.0 + a.X)$ and $Q = a.0 + fix(X = a.X)$ then $\langle\langle P\rangle\rangle$ consists of infinitely many trees while $\langle\langle Q\rangle\rangle$ has two trees as shown below.

As in the finite case one obtains a refinement notion as inclusion between sets of the possible worlds.

Definition 4. Let $P, Q \in \mathbf{RBP}$ be closed processes. Q is a *possible worlds refinement* of P, written $P \leqslant_D Q$ iff $\langle\langle Q\rangle\rangle \subseteq \langle\langle P\rangle\rangle$. P and Q are possible worlds equivalent, $P =_D Q$, iff $\langle\langle P\rangle\rangle = \langle\langle Q\rangle\rangle$.

Example 3. $P \leqslant_D Q$ where P, Q are taken from Example 2.



5. Properties of the infinite possible worlds refinement

Veglioni and De Nicola [10] showed that bisimulation implies possible worlds equivalence for finite processes. We show first that this result carries over to infinite processes.

The following technical lemma is needed for arguments involving recursive processes.

Lemma 5. *Let* $P, P_1, \ldots, P_n \in \mathbf{RBP}$ *and* $X_1, \ldots, X_n \in Idf$, $X_i \neq X_j$ *for* $i \neq j$, $\sigma \in \mathbf{ENV}$. *Then*

$$\langle\langle P[X_1/P_1, \ldots, X_n/P_n]\rangle\rangle(\sigma) = \langle\langle P\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)].$$

Proof. By induction on the structure of P. We only show the case

$$P = fix(X = P'). \qquad \square$$

We assume w.l.o.g. that X does not appear free in $P_1, \ldots, P_n$.

Case 1: $X \notin \{X_1, \ldots, X_n\}$. Then

$$P[X_1/P_1, \ldots, X_n/P_n] = fix(X = P'[X_1/P_1, \ldots, X_n/P_n]).$$

By induction hypothesis

$$\langle\langle P'[X_1/P_1, \ldots, X_n/P_n]\rangle\rangle(\sigma) = \langle\langle P'\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)]$$

hence for all T

$$\Phi_{P'[X_1/P_1,\ldots,X_n/P_n],X}(\sigma)(T) = \Phi_{P',X}(\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)](T)$$

hence

$$\begin{aligned}
&\langle\langle P[X_1/P_1, \ldots, X_n/P_n]\rangle\rangle(\sigma) \\
&\quad = fix\ \Phi_{P'[X_1/P_1,\ldots,X_n/P_n],X}(\sigma) \\
&\quad = fix\ \Phi_{P',X}(\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)] \\
&\quad = \langle\langle P\rangle\rangle\sigma[X_1/\langle\langle P_1\rangle\rangle(\sigma), \ldots, X_n/\langle\langle P_n\rangle\rangle(\sigma)].
\end{aligned}$$

Case 2. $X = X_i$ for some i: similarly

To establish the relation between the two equivalence notions we use auxiliary functions h_i, $i = 1, 2$, defined in the following.

Definition 5. Let $D_1 = \mathbf{D}$, $D_2 = \mathbf{T}_{\text{finbran}}$ and $\mathbf{Env}_i = \{\sigma : Idf \to D_i\}$.
$h_i : \mathbf{RBP} \to \mathbf{Env}_i \to D_i$ is given by
$h_i(0)(\sigma) = \emptyset$,

$h_i(X)(\sigma) = \sigma(X)$,

$h_i(a.P)(\sigma) = a \cdot h_i(P)(\sigma)$,

$h_i(P_1 + P_2)(\sigma) = h_i(P_1)(\sigma) + h_i(P_2)(\sigma)$,

$h_i(fix(X = P))(\sigma) = fix\ \varphi_{i,P,X}(\sigma)$,

where $fix\ \varphi_{i,P,X}(\sigma)$ is the unique fixed point of the contractive mapping $\varphi_{i,P,X}(\sigma): D_i \to D_i$ where $\varphi_{i,P,X}(\sigma)(t) = h_i(P)\sigma[X/t]$, $i = 1, 2$.

Lemma 6. *Let P be a deterministic process, $\sigma \in \mathbf{Env}_1$*

$$\langle\langle P\rangle\rangle(\tilde{\sigma}) = \{h_1(P)(\sigma)\},$$

where $\tilde{\sigma}(X) = \{\sigma(X)\}$ for all $X \in Idf$. In particular, $\langle\langle P\rangle\rangle = \{h_1(P)\}$ for all closed deterministic P.

Proof. By induction on the structure of P. The basis of induction and the handling of the operators choice and prefixing are straightforward. For the case $P = fix(X = P')$ it is easily shown that $\{h_1(P)(\sigma)\}$ is a fixed point of $\Phi_{P',X}(\tilde{\sigma})(T)$.

Remark 5. Let $P, P_1, \ldots, P_n \in \mathbf{RBP}$ and $X_1, \ldots, X_n \in Idf$, $X_i \neq X_j$, for $i \neq j$. Then

$$h_2(P[X_1/h_2(P_1), \ldots, X_n/h_2(P_n)])(\sigma)$$

$$= h_2(P)\sigma[X_1/h_2(P_1)(\sigma), \ldots, X_n/h_2(P_n)(\sigma)]$$

holds in analogy to Lemma 5.

Let $F: \mathbf{T}_{\text{finbran}}/_{\sim} \to \mathbf{D}$ be given as in Theorem 3.

Lemma 7. *Let $P \in \mathbf{RBP}$ and $X_1, X_2, \ldots, X_n \in Idf$ be the identifiers that occur free in P and $P_1, \ldots, P_n$ closed processes $\in \mathbf{RBP}$. Let $\tau \in \mathbf{Env}_2$ with $\tau(X_i) = h_2(P_i)$ and $\sigma \in \mathbf{Env}_1$ with $\sigma(X_i) = F([h_2(P_i)]_{\sim})$. Then $h_1(P)\sigma = F([h_2(P)\tau]_{\sim})$*

Proof. By induction on the structure of P. The basis of induction is obvious.
Induction step:

1. $P = a.P'$:

$$\begin{aligned} h_1(P)(\sigma) &= \{(a, h_1(P')(\sigma))\} = \{(a, F[h_2(P')(\tau)]_{\sim})\} \\ &= \lim\{(a, F_{n-1}([(h_2(P')(\tau))^{(n-1)}]_{\sim}))\} \\ &= \lim F_n([(a \cdot h_2(P')(\tau))^{(n)}]_{\sim}) \\ &= F([a \cdot h_2(P')(\tau)]_{\sim}) \\ &= F([h_2(a.P')(\tau)]_{\sim}) = F([h_2(P)(\tau)]_{\sim}). \end{aligned}$$

2. $P = P_1 + P_2$: by simple calculation.

3. $P = fix(X = P')$: we show that $F([h_2(fix(X = P'))(\tau)]_\sim)$ is a fixed point of $\varphi_{1,P',X}(\sigma)$.

$$\begin{aligned} F([h_2(fix(X = P'))(\tau)]_\sim) &= F([h_2(P')\tau[X/h_2(P)(\tau)]]_\sim) \\ &= F([h_2(P')\tau[X/h_2(P[X_1/P_1,\ldots,X_n/P_n])]]_\sim) \\ &= F([h_2(P')(\tau')]_\sim) = h_1(P')(\sigma') \\ &= h_1(P')\sigma[X/F([h_2(P[X_1/P_1,\ldots,X_n/P_n])]_\sim)] \\ &= h_1(P')\sigma[X/F([h_2(P)(\tau)]_\sim)] \end{aligned}$$

by induction assumption, where $\sigma'(X) = F([h_2(P[X_1/P_1,\ldots,X_n/P_n])]_\sim)$ and $\sigma'(Y) = \sigma(Y)$ for $Y \neq X$.

Lemma 8. *Let* $P \in \mathbf{RBP}$, $a \in \mathbf{Act}$, $\sigma \in \mathbf{Env}_2$. *If* $P \xrightarrow{a} P'$ *then* $h_2(P)(\sigma) \xrightarrow{a} h_2(P')(\sigma)$.

Proof. By structural induction. We only show the case

$$P = fix(X = Q).$$

Let $P \xrightarrow{a} P'$, i.e. $Q[X/P] \xrightarrow{a} P'$. By Lemma 9.3 in [2] there exists $Q' \in \mathbf{RBP}$ such that $Q \xrightarrow{a} Q'$ and $Q'[X/P] = P'$. We apply the induction hypothesis to Q; hence,

$$h_2(Q)(\sigma) \xrightarrow{a} h_2(Q')(\sigma) \quad \text{for all } \sigma,$$

hence,

$$h_2(Q)\sigma[X/h_2(P)(\sigma)] \xrightarrow{a} h_2(Q')\sigma[X/h_2(P)(\sigma)] \quad \text{for all } \sigma,$$

hence, $h_2(P)(\sigma) \xrightarrow{a} h_2(P')(\sigma)$ by Remark 5.

Lemma 9. *Let* $P \in \mathbf{RBP}$ *and* $X_1,\ldots,X_n$ *be the identifiers that occur free in* P, $X_i \neq X_j$, *for* $i \neq j$. *Let* $\sigma \in \mathbf{Env}_2$ *such that* $\sigma(X_i) = P_i$ *where* $P_i \in \mathbf{RBP}$ *is closed. If* $X_1,\ldots,X_n$ *are guarded in* P *then for all* $t' \in \mathbf{T}_{\text{finbran}}$ *if* $h_2(P)\sigma \xrightarrow{a} t'$ *then there exists* $P' \in \mathbf{RBP}$, P' *closed, such that* $P[X_1/P_1,\ldots,X_n/P_n] \xrightarrow{a} P'$ *and* $h_2(P') = t'$.

Proof. By structural induction, Lemma 9.3 in [2] and Remark 5. □

Lemma 10. $R = \{(Q, h_2(Q)) \mid Q \in \mathbf{RBP}$ *closed*$\}$ *is a bisimulation between* $(\mathbf{RBP}, \mathbf{Act}, \rightarrow, P)$ *and* $(\mathbf{T}_{\text{finbran}}, \mathbf{Act}, \rightarrow, h_2(P))$.

Proof. By Lemmas 8 and 9.

Definition 6. Let $t \in \bigcup_{i \geqslant 0} D_i$, $I(t) = \{a_1, \ldots, a_n\}$.

$$t(a) := \{t' \mid (a, t') \in t\} \quad a \in \mathbf{Act}$$

$det(t) := \{\{(a_1, x_1), \ldots, (a_n, x_n)\}$ where $x_i \in det(t')$, $t' \in t(a_i)$ for $i = 1, \ldots, n\}$

Lemma 11. *$det(t)$ is non-distance-increasing on $\bigcup_{i \geqslant 0} D_i$.*

Proof. By Lemma 2.

Hence det can be extended to $det : \mathbf{D} \to \mathbf{D}$.

Lemma 12. *Let $P \in \mathbf{RBP}$, $\sigma \in \mathbf{Env}_1$, $\tilde{\tilde{\sigma}}(X) = det(\sigma(X))$. Then*

$$\langle\langle P \rangle\rangle(\tilde{\tilde{\sigma}}) = det(h_1(P)(\sigma)).$$

In particular, $\langle\langle P \rangle\rangle = det(h_1(P))$ for all closed processes P.

Proof. By induction on the structure of P. We consider the case

$$P = fix(X = P')$$

and show that $det(h_1(P)(\sigma)) = det(h_1(fix(X = P'))(\sigma))$ is a fixed point of $\Phi_{P',X}(\tilde{\tilde{\sigma}})$.

$$\begin{aligned} det(h_1(P)(\sigma)) &= det(h_1(P'))\sigma[X/h_1(P)(\sigma)]) \\ &= det(h_1(P')(\sigma')) \\ &= \langle\langle P' \rangle\rangle(\tilde{\tilde{\sigma}}) \end{aligned}$$

by induction hypothesis, where

$$\sigma'(Y) = \sigma(Y) \quad \text{for } Y \neq X, \quad \sigma'(X) = \langle P \rangle(\sigma),$$

hence,

$$det(h_1(P)(\sigma)) = \langle\langle P' \rangle\rangle \tilde{\tilde{\sigma}}[X/det(h_1(P)(\sigma)];$$

hence $det(h_1(P)(\sigma))$ is a fixed point of $\Phi_{P',X}(\tilde{\tilde{\sigma}})$ and $\langle\langle P \rangle\rangle(\tilde{\tilde{\sigma}}) = det(h_1(P)(\sigma))$. □

Theorem 6. *Let $P, Q \in \mathbf{RBP}$ be closed processes. If $P \sim Q$ then $P =_D Q$,*

Proof. By Lemmas 7, 10 and 12. □

Veglioni and De Nicola [10] gave an axiomatization of the possible world equivalence for finite processes. It is easy to see that these axioms hold also for infinite processes. In addition, the possible worlds refinement satisfies the expected law for recursive processes. The question of an axiomatization of possible world refinement in the general case is still open.

Lemma 13. *Let $P' \in \mathbf{RBP}$ and X be the only variable occurring free in P'.* $fix(X = P') =_D P'[X/fix(X = P')]$.

Proof. As $\langle\langle P'[X/fix(X = P')]\rangle\rangle$ is a fixed point of $\Phi_{P',X}(\sigma)(T) = \langle\langle P'\rangle\rangle\sigma[X/T]$ by Lemma 5.

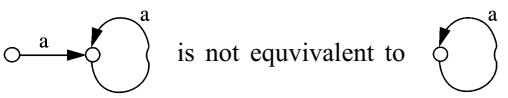
6. Extensions, related work and open problems

6.1. General underspecification

It might be desirable to be able to describe underspecification that allows to combine arbitrary processes to an underspecified term. Veglioni and De Nicola [10] introduce for this purpose an operator $\oplus$ that is used in combination with $+$ interpreted as above. So e.g. $(a.P + a.Q) \oplus c.R$ displays underspecification twice on the top level. Its meaning is a set consisting of three trees, provided P, Q and R are deterministic. One problem with this view arises if we want to introduce a parallel construct into the language. If we maintain that the meaning of a process should be a set of deterministic trees then the meaning of $a.b.0|a.c.0$ has to be a set consisting of two trees. In a refinement step one of them could be discarded which means that certain computation paths would be lost. On the other hand, the meaning of $a.b.0|c.d.0$ is a singleton set containing all information and there is no further refinement. This problem can be solved by admitting sets of nondeterministic trees as semantics for a language including a parallel construct. It should be noted that $\mathscr{P}_{nco}(\mathbf{D})$ is still a suitable domain for such an interpretation. However, the definition of the operation $*$ has to be modified. Another way to incorporate underspecification is to separate the issues of nondeterminism and underspecification completely by introducing a seperate underspecification operator and interpreting $+$ in the standard way. In this setting a parallel construct can be easily incorporated and again $\mathscr{P}_{nco}(\mathbf{D})$ is still a suitable domain, compare [5].

6.2. Operational characterization

Veglioni and De Nicola [10] give a characterization of the possible worlds equivalence in terms of the standard operational semantics of **BP** by decomposing the transition graph of a process Q into a set $PW(Q)$ of deterministic graphs, see [10], and later argue that this characterization does not carry over to the infinite case as their definition is not sufficiently abstract for loops, as e.g.



Veglioni and De Nicola [10] propose to solve this problem by using a graph equivalence that is weaker than graph isomorphism such that the above two transition systems will become equivalent.

Let us assume that we choose such a weak notion of equivalence that identifies the above graphs. Then the resulting operational semantics will be incomparable with the denotational semantics in the sense that

(i) there are processes P and Q for which $PW(P)=PW(Q)$ but $P\neq_D Q$
(ii) there are processes P and Q for which $P=_D Q$ but $PW(P)\neq PW(Q)$.

Example 4. Let $P=fix(X=a.0+a.X)$ and $Q = a.0+fix(X=a.X)$. Under the above assumption the operational meanings $PW(P)$ and $PW(Q)$ (in the sense of Veglioni and De Nicola [10]) consist both of two graphs, i.e. one branch labelled a and one loop labelled a, but $Q\neq_D P$.

Example 5. Let $P=fix(X=R)=fix(X=a.0+a.X)$ and $Q=R[X/fix(X=R)]=a.0+a.fix(X=a.0+a.X)$, then $P=_D Q$, but $PW(P)\neq PW(Q)$ under the assumed notion of graph equivalences.

The question of an appropriate operational semantics for **RBP** remains open.

6.3. Concatenation and infinite sums

It is not difficult to see that concatenation can be easily incorporated into our setting. We omit the 0 process, consider instead each $a\in$ **Act** as a basic process and substitute prefixing by concatenation. The corresponding semantic operator on $\bigcup_{i\geqslant 0}\mathbf{D}_i$ is non-distance-increasing and hence all constructions carry over to this case.

6.4. Infinite sums

For simplicity, we considered in **RBP** a standard binary 'choice'. It should be noted that the approach can be extended to an infinite summation operator $\sum$. However, we then no longer choose **D** as our basic domain, but the complete pseudometric space $\mathbf{T}/_{\sim}$, as **D** is too coarse to distinguish between $P_1=\sum_{i\geqslant 1} a^i$ and $P_2=\sum_{i\geqslant 1} a^i + fix(X=a.X)$. Here **T** denotes the class of (isomorphism classes of) trees with labels in **Act**.

6.5. Related work

The idea of two different types of nondeterminism and their modelling by branching, respectively, sets is not new. In the special case of a *finite* alphabet **Act** it can be found in [8], where a CSP-type language with the choice-operator of Dijkstra and the $\Box$-operator of Brookes et al. [1] is considered and interpreted in terms of (sets of) trees. For a finite alphabet **Act** [8] establishes the relation between the Hennessy–Milner logic HML and $\mathscr{P}_c(\mathbf{T})$ where $\mathscr{P}_c(\mathbf{T})$ denotes the closed subsets of the pseudometric space **T**.

References

[1] S.D. Brookes, C.A.R. Hoare, A.W. Roscoe, A theory of communicating sequential processes, J. ACM 31 (3) (1984) 560–599.
[2] C. Baier, M.E. Majster-Cederbaum, The connection between an event structure semantics and an operational semantics for TCSP, Acta Inform. 31 (1) (1994) 81–104.
[3] J.W. de Bakker, J.I. Zucker, Processes and the denotational semantics of concurrency, Inform. and Control 54 (1/2) (1982) 70–120.
[4] R. Engelking, General Topology, Heldermann Verlag, 1989.
[5] M.E. Majster-Cederbaum, Underspecification for process algebras, Technical report TR7/2000, Fakultät für Mathematik und Informatik, Universität Mannheim, 2000.
[6] M.E. Majster-Cederbaum, F. Zetzsche, Towards a foundation for semantics in complete metric spaces, Inform. and Comput. 90 (2) (1991) 217–243.
[7] M. Nivat, Infinite words, infinite trees, infinite computations, Math. Centre Tracts 109 (1979) 1–52.
[8] W.C. Rounds, On the relationship between Scott domains, synchronization trees, and metric spaces, Inform. and Control 66 (1/2) (1985) 6–28.
[9] R.J. van Glabbeek, The linear time–branching time spectrum, in: J.C.M. Baeten, J.W. Klop (Eds.), Proceedings of the CONCUR 90, Amsterdam, Lecture Notes in Computer Science, vol. 458, Springer, Berlin, 1990, pp. 278–297.
[10] S. Veglioni, R. De Nicola, Possible Worlds for Process Algebras, Lecture Notes in Computer Science vol. 1466, Springer, Berlin, 1998, pp. 179–193.